User Authentication for Role-Based Access Control

The user authentication system is a part of Role-Based Access Control (RBAC) project for the high-level LHC Control Software (LSA) at CERN. The project is being developed by LAFS ("LHC at Fermilab Software") collaboration between control groups at CERN and Fermilab. The function of RBAC authentication is to create, distribute, and manage digital credentials for the users. We had to consider many constraints dictated by the existing control system, and diversity of the used software.

In the design of RBAC we separate the concepts of *authentication* (A1) and *authorization* (A2).

The purpose of the RBAC authentication system is to verify the digital identity of a principal (which is either a human user or a program). This is accomplished in several ways:

- By verifying the user name and password.
- By verifying a client-side X.509 certificate.
- By the client's network address.
- By using an existing authentication token.

In any case, if the authentication succeeds its result is a digitally signed authentication token that is returned to the application. The program can use the token whenever it needs to interact with various parts to the control system. For example, the token can be provided as one of the arguments in an RMI call to set a device. Front-ends and the middleware that are receiving such calls will verify the token, thus confirming the identity of the remote party, and can use it as a base for authorization

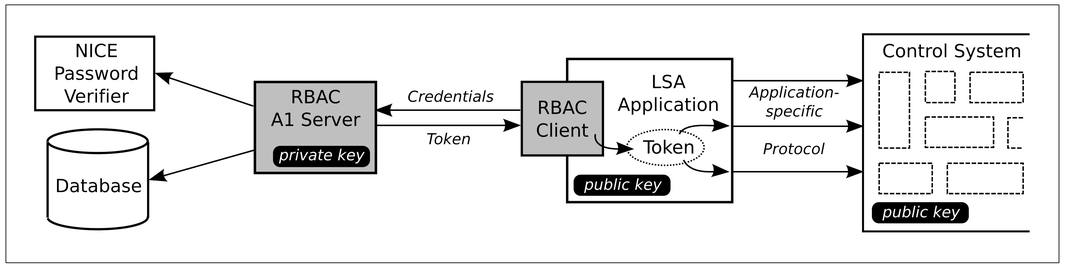
RBAC Authentication Token

Authentication Data	Serial Number
	Authentication Time
	Expiration Time
	Application Descriptor
	Application Timeout
	Location Address
	User Name
	List of Roles
Auxiliary Data *(used internally by A1 client)*	
Digital Signature	

Main features:

- None of the data is encrypted.
- The token has a standardized binary format, so that it can be implemented in various programming languages in platforms.
- The token can be validated quickly and easily inside the middleware and front-ends, without contacting the server.
- To validate a token, the program should verify its digital signature and check the expiration time.

Technical Design



The RBAC Authentication System is build in a common client/server model.

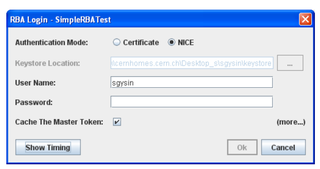
- A1 server receives authentication requests via HTTP from multiple clients, returning back either an authentication token, or an error code. Implemented as a servlet running on Jetty 6. Several servers are used in parallel for performance and reliability.
- The client side is organized as a library, built in other applications or application frameworks. It provides a function to be called in order to obtain the authentication token from the server, and several standard GUI components. The client is implemented in Java and C++.
- Two services on the back end: the database and a NICE password verifier. *(NICE is a site-wide user account database at CERN)*. Users' passwords are not stored in the RBAC database.

RBAC uses one principal keypair,

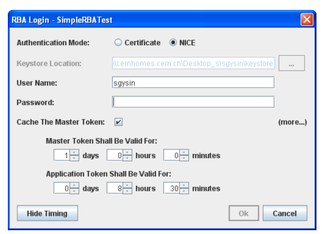
- The private key is known only to A1 server. Used to sign outgoing tokens and for server-side authentication in SSL/TLS.
- The public key is known for all applications and front-ends (distributed inside the programs' .jar files). Used to validate the tokens, and to verify the server's identity before sending user credentials.

Additional functions of the RBAC client are Single-Sign On and a "Role Picker" (to reduce the number of roles a powerful user possesses).

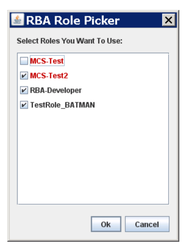
Examples of Default User Interfaces



Basic Login Dialog



Extended Login Dialog



Role Picker

ICALEPCS'07
TPPA12

Andrey D. Petrov, Suzanne Gysin, and Carl Schumann
Fermi National Accelerator Laboratory, Batavia, IL 60510, U.S.A.

Fermilab

CERN